# Enterprise Architecture Strategies
# Conceptual Architecture Principles

State of CT Architecture Team
Revised: 8/22/2000

# Business Oriented

## Information Is an Enterprise Asset

**Principle 1.** Information is valued as an enterprise asset, which must be shared to enhance and accelerate decision making.

### Justification

- Decision-making requires information beyond the traditional borders of a system or agency.
- Enables new enterprise-wide or multi-agency solutions.
- Enhances the efficiency and effectiveness of the delivery of services.
- Most information is in isolated pockets such that the value of information is not always recognized.
- Treating the data as an enterprise asset increases its integrity and the relevance of data.

### Implications

- Need to develop policy pertaining to information stewardship.
- Information value must be identified, authenticated, and leveraged.
- Need for unified information management.
- Need to establish supporting policies for security, privacy, confidentiality and information sharing.
- Data needs to be structured for easy access and management.

## Architecture Management

**Principle 2.** The planning and management of the State's enterprise-wide technical architecture *must be unified* and *have a planned evolution that is governed across the enterprise.*

### Justification

- Without a unified approach, there will be multiple, and possibly conflicting, architectures.
- Good change requires collaboration and collective planning.
- Architecture must be well thought out.
- Governance will be simplified.

### Implications

- A unified approach will require a change in cultural attributes.
- Normal evolution will require prioritization and reprioritization across all IT initiatives.
- Dependencies must be maintained.
- The architecture must be continually re-examined and refreshed.
- Short-term results vs. long term impact must be constantly considered.
- Establishing enterprise architecture takes time and involves a lot of change.

## Architecture Compliance

**Principle 3.** Architecture support and review structures shall be used to ensure that the integrity of the architecture is maintained as systems and infrastructure are acquired, developed and enhanced.

### Justification

- To realize the benefits of a standards-based enterprise architecture, all information technology investments must ensure compliance with the established IT architecture.
- For maximum impact, review should begin as early in the solution planning process as possible
- "If you are going to talk the talk, then you must be willing to walk the walk."

### Implications

- A structured project level review process will be needed to ensure that information systems comply with the IT Architecture and related standards.
- Processes incorporating the principles of this (technical) architecture must be developed for all application procurement, development, design, and management activities.
- This compliance process must allow for the introduction of new technology and standards.
- Conceptual Architecture and Technical Domain principles should be used as evaluation criteria for purchasing as well as developing software.

## Leverage Data Warehouses

**Principle 4.** We should leverage data warehouses to facilitate the sharing of existing information to accelerate and improve decision-making at all levels.

### Justification

- Data can be replicated and combined from multiple agencies without changing the originating systems or developing new systems.
- Reduced business cycle times have led to a need for faster access to more information.
- There is a significant burden on programmers to generate reports and data queries. Data warehouses and their associated end-user tools make it possible to relieve this burden by making it the responsibility of end users.
- Warehouses fulfill the need for internally consistent data.

### Implications

- Data warehousing must become a core competency of IT.
- Data warehouses become types of configuration standards that need to be developed and maintained.
- End-user tools must be provided.
- End users become more knowledgeable about information and the tools they need to access and analyze it.
- The processes and procedures for data extraction, "cleansing", and the loading of warehouses will require high levels of reliability and integrity.

## Ensure Security, Confidentiality and Privacy

**Principle 5.** IT systems should be implemented in adherence with all security, confidentiality and privacy policies and applicable statutes.

### Justification

- Helps to safeguard confidential and proprietary information
- Enhances public trust
- Enhances the proper stewardship over public information
- Helps to ensure the integrity of the information

### Implications

- Need to identify, publish and keep the applicable policies current.
- Need to monitor compliance to policies.
- Must make the requirements for security, confidentiality and privacy clear to everyone.
- Education on issues of privacy and confidentiality must become a routine part of normal business processes.

## Reduce Integration Complexity

**Principle 6.** The enterprise architecture must reduce integration complexity to the greatest extent possible.

### Justification

- Increases the ability of the enterprise to adapt and change.
- Reduces product and support costs

### Implications

- Decreases the number of vendors, products, and configurations in the State's environment.
- Must maintain configuration discipline.
- Will sacrifice performance and functionality in some instances.
- Will rely on components supplied by vendors.
- Determination of "the greatest extent possible" includes consideration of how reducing complexity can negatively impact providing critical client services.

## Re-use before Buying, Buy before Building

**Principle 7.** We will consider re-use of existing applications, systems, and infrastructure before investing in new solutions. We will build only those applications or systems that will provide clear business advantages and demonstrable cost savings

### Justification

- Use and availability of effective packaged solutions is increasing.
- Using tested solutions reduces risks.
- Reduces the total cost of ownership.

### Implications

- Software license agreements and system development contracts should be written to allow for re-use across State government.
- "The definition of "reusable" will include solutions available from other government entities (e.g., other states, federal government, etc.).
- Areas that provide clear advantages and businesses cost savings are likely to require quick adaptation.
- Must identify the areas in which the State is seeking to distinguish itself.

## Integration

**Principle 8.** Systems must be designed, acquired, developed, or enhanced such that data and processes can be shared and integrated across the enterprise and with our partners.

### Justification

- Increase efficiency while better serving our customers (e.g., the public, agencies, etc.).
- Redundant systems cause higher support costs.
- Ensures more accurate information, with a more familiar look and feel.
- Integration leads to better decision making and accountability.

### Implication

- IT staff will need to consider the impacts on an enterprise wide scale when designing applications.
- We will need new tools and training for their proper use.
- Will need a method for identifying data and processes that need integration, when integration should take place, whom should have access to the data, and cost justification for integration.
- Will need a "coordinator" that can maintain and arbitrate a common set of domain tables, data definitions, and processes across the organization.
- Over integration can lead to difficult data management and inefficient processes.

## Reengineer First

**Principle 9.** New information systems will be implemented after business processes have been analyzed, simplified or otherwise redesigned as appropriate.

### Justification

- There is no real "value" in applying technology to old, inefficient processes.
- Work processes will be more streamlined efficient and cost effective.
- Work processes, activities, and associated business rules will be well understood and documented.
- Reduces the total cost of ownership.

### Implications

- Need to have an agreed upon business re-engineering process.
- New technology will be applied in conjunction with business process review.
- Business processes must be optimized to align with business drivers.
- Additional time and resources will have to be invested in analysis early in the systems life cycle.
- Organizational change will be required to implement reengineered work processes.
- May require regulatory or legislative change.

## Total Cost of Ownership

**Principle 10.** Adopt a total cost of ownership model for applications and technologies which balances the costs of development, support, training, disaster recovery and retirement against the costs of flexibility, scalability, ease of use, and reduction of integration complexity.

### Justification

- Consideration of all costs associated with a system over its entire life span will result in significantly more cost effective system choices.
- Enables improved planning and budget decision-making.
- Reduces the IT skills required for support of obsolete systems or old standards.
- Simplifies the IT environment.
- Leads to higher quality solutions.

### Implications

- The State budget process needs to accommodate Total Cost of Ownership of a system over a longer timeframe than current budgeting models.
- Will require looking closely at technical and user training costs especially when making platform or major software upgrades during the lifetime of the system.
- Requires designers and developers to take a systemic view.
- Need to selectively sub-optimize individual IT components.
- Need to develop a cost of ownership model.
- Need to ensure coordinated retirements of systems.

## Minimize Platform Configurations

**Principle 11.** Create a small number of consistent configurations for deployment across the enterprise.

### Justification

- Reducing uniqueness in product selection and standardization reduces support and maintenance costs, and simplifies training and skills transfer.
- The cost of IT personnel is increasing and the cost of hardware is decreasing rapidly.
- This is the most efficient approach to enterprise-wide infrastructure configuration and maintenance.
- By constantly 'tweaking' the performance of an individual server or desktop computer, a multitude of unique configurations is created, thus increasing support and maintenance costs.

### Implications

- Increased initial capital investment.
- Deploy applications on uniformly configured servers ("If in doubt, use the bigger Box").
- Plan to replace multiple, non-standard, configurations with a small number of consistent configurations.
- Plan for the regular replacement of platform components to ensure the retirement of obsolete and unique configurations.

## Basic Information Services

**Principle 12.** A standardized set of basic information services (e.g., email, voicemail, e-forms, user training) will be provided to all employees.

### Justification

- Increases productivity.
- Reduces costs of maintenance.
- Provides the basis for multi-agency or statewide business initiatives.
- Provides for universal employee access to information.

### Implications

- Basic services definition needs to be created and regularly reviewed.
- May increase "one-time" costs to upgrade to the minimum service level.
- Training must be provided to all users of basic services.

# Technology Oriented

## Shared Components Using an N-Tier Model

**Principle 13.** Applications, systems and infrastructure will employ reusable components across the enterprise, using an *n-tier* model.

### Justification

- You can make significant changes to a component of a system, such as changing from a Windows client to a web-browser client, without changing the rest of the system.
- Enables simplification of the environment and geographical independence of servers.
- Takes advantage of modular off-the-shelf components
- Reuse will lower costs and maintenance efforts.
- Allows for leveraging skills across the enterprise.

### Implications

- Component management must become a core competency.
- Requires the development a culture of reuse.
- Reusable components must be platform independent.
- Physical configuration standards must be established.
- Design reviews become crucial.
- Application systems must be highly modularized without making components too small or too simple to do useful “work”.

## Logical Partitioning and Boundaries

**Principle 14.** The logical design of application systems and databases should be highly partitioned. These partitions must have *logical boundaries* established, and the logical boundaries *must not be violated*.

### Justification

- A change in a database or application can potentially affect many large programs, if they are not highly partitioned.
- You can not separate the components in a system from each other without creating logical boundaries.
- Recoding leads to time-consuming re-testing.
- Partitioning isolates/minimizes change impact.
- Partitioned code is more adaptive to changes in internal logic, platforms, and structures.

### Implications

- Applications need to be divided into coded entities (e.g., presentation, process, and data access).
- For databases, there will be a need to develop competency in partitioning horizontally and vertically; this will result in more but simpler tables.
- Design reviews must ensure logical boundaries are kept intact.

## Message-Based Interfaces

**Principle 15.** The interfaces between separate application systems must be *message-based*; this applies to both internal and external systems.

### Justification

- The use of messaging is important for enforcing the architecture principle of *logical partitioning and boundaries*.
- Enables rapid response in maintenance and enhancement activities as required by changes in business processes.
- Messaging technology simplifies integration efforts.
- Messaging technology allows for transparency in locations, databases, and data structures.

### Implications

- The implementation of a messaging infrastructure will be necessary.
- Trust, or letting go of control, is often the most difficult aspect among IT staff as messaging is introduced into the IT culture.
- Common messaging formats, IDs, and standards must be established.
- Developers must learn how to use messaging.
- Network traffic will increase.

## Event-Driven Systems

We must deploy application systems that are driven by *business events*.

### Justification

- Enables applications to adapt quickly to changes in business processes by only changing the application component related to the changed business event.
- Strengthens linkage to the business by mirroring the actual business environment.
- Easier to realign IT when change occurs.

### Implications

- Requires systemic thinking as event-based processing crosses traditional system boundaries.
- Business processes need to be optimized to obtain full benefits.
- Need to retrain developers to incorporate business concepts in their software development methods.

## Physical Partitioning of Processing

**Principle 16.** We should separate on-line transaction processing (OLTP) from data warehouse and other end-user computing.

### Justification

- Separating end-user requests and OLTP maximizes the efficiency of both environments.
  - Growth in OLTP is incremental, and requirements are predictable.
  - Growth in data warehouses and end-user computing has been nonlinear, and requirements are very difficult to predict.
- Fosters the concept of data stewardship.

### Implications

- Data warehouse become types of configuration standards that need to be developed and maintained.
- Data warehousing becomes a core competency of IT.
- Business and IT must agree on the purpose and objective of the data warehouses.
- Business users must justify the cost of compiling and maintaining the data (warehouse).
- Operational systems are candidates for outsourcing all or parts — development or maintenance.

## Formal Software Engineering

**Principle 17.** The State shall adopt and employ consistent software engineering practices and methods based on accepted industry standards.

### Justification

- Reduces training costs.
- Leads to benchmarks for measurement.
- Enables improved quality assurance.
- Facilitates the reuse of programming modules and code.

**Principle 18.** Implications

- Need to agree on practices and methods.
- Requires training in the practices and methods.
- Requires monitoring for compliance.
- All State IT organizations and third party developers will employ the State's software engineering practices and methods.
- Need to develop in-house software engineers.

# Business Continuity Oriented

## Mainstream Technologies

**Principle 19.** IT solutions will use industry-proven, mainstream technologies.

### Justification

- Avoids dependence on weak vendors.
- Reduces risk.
- Ensures robust product support.
- Enables greater use of commercial-off-the-shelf solutions.

### Implications

- Need to establish criteria for vendor selection and performance measurement.
- Need to establish criteria to identify the weak vendors and poor technology solutions.
- Requires migration away from existing weak products in the technology portfolio

## Industry Standards

**Principle 20.** Priority will be given to products adhering to industry standards and open architecture.

### Justification

- Avoids dependence on weak vendors.
- Reduces risks.
- Ensures robust product support.
- Enables greater use of Commercial-off-the-Shelf solutions.
- Allows flexibility and adaptability in product replacement.

### Implications

- Requires a culture shift.
- Need to establish criteria to identify standards and the products using them.
- IT organizations will need to determine how they will transition to this mode.

## Disaster Recovery / Business Continuity

**Principle 21.** An assessment of business recovery requirements is mandatory when acquiring, developing, enhancing or outsourcing systems. Based on that assessment, appropriate disaster recovery and business continuity planning, design and testing will take place.

### Justification

- Due to factors such as the Internet and Y2K, customers and partners have heightened awareness of systems availability.
- The pressure to maintain availability will increase in importance. Any significant visible loss of system stability could negatively impact our image.
- Continuation of business activities without IT is becoming harder.
- Application systems and data are valuable State assets that must be protected.

### Implications

- Systems will need to be categorized according to business recovery needs (e.g. business critical, non-critical, not required).
- Alternate computing capabilities need to be in place.
- Systems should be designed with fault tolerance and recovery in mind.
- Plans for work site recovery will need to be in place.
- Costs may be higher.

## Enterprise Network as Virtual LAN

**Principle 22.** We must implement a statewide backbone network that provides a virtual, enterprise-wide local area network.

### Justification

- Networks are the essential enabling technology for client/server, Internet, and collaborative computing.
- This is the basis of the anywhere, anytime, seamless access that is a major driver of agencies.
- Knowledge workers' increasing need for access to information across the enterprise.
- Lack of a robust network architecture will impact the success of distributed applications.
- Expands the vision of organizations by reaching out to customers and suppliers.

### Implications

- Need to implement a robust, unified directory services capability.
- Requires higher speed networks and higher bandwidth networks.
- Interconnection of distributed LANs.
- Need to create connections from legacy systems to client/server and Internet applications.

## Scalability

**Principle 23.** The underlying technology infrastructure and applications must be scalable in size, capacity, and functionality to meet changing business and technical requirements.

### Justification

- Reduces Total Cost of Ownership by reducing the amount of application and platform changes needed to respond to increasing or decreasing demand on the system.
- Encourages reuse.
- Leverages the continuing decline in hardware costs.

### Implications

- Scalability must be reviewed for both "upward" and "downward" capability.
- May increase initial costs of development and deployment.
- Will reduce some solution choices.